Triangle Attack: A Query-efficient Decision-based Adversarial Attack

Xiaosen Wang[1,2], Zeliang Zhang[1], Kangheng Tong[1], Dihong Gong[2], Kun He[1], Zhifeng Li[2], Wei Liu[2]

[1] School of Computer Science, Huazhong University of Science and Technology

[2] Data Platform, Tencent

Contact: xiaosen@hust.edu.cn

Homepage: http://xiaosenwang.com/

11/17/2022

Adversarial Example

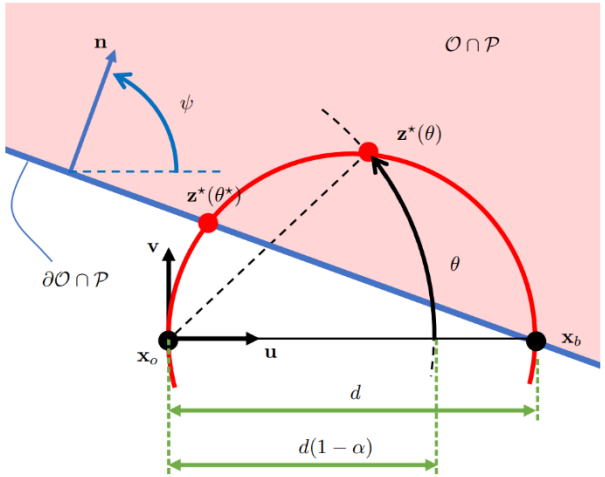
Adversarial examples are **indistinguishable** from legitimate ones by adding small perturbations, but lead to **incorrect model prediction**.

Decision-Based Attack: Attacker can only access the **prediction (top-1) label** of the victim model, which is more applicable in **real-world** scenarios.

Background: Existing attacks (*e.g.*, BA, OPT, etc.) have exhibited great effectiveness, but using **thousands of queries**, which plays a significant role when deployed in real-world.

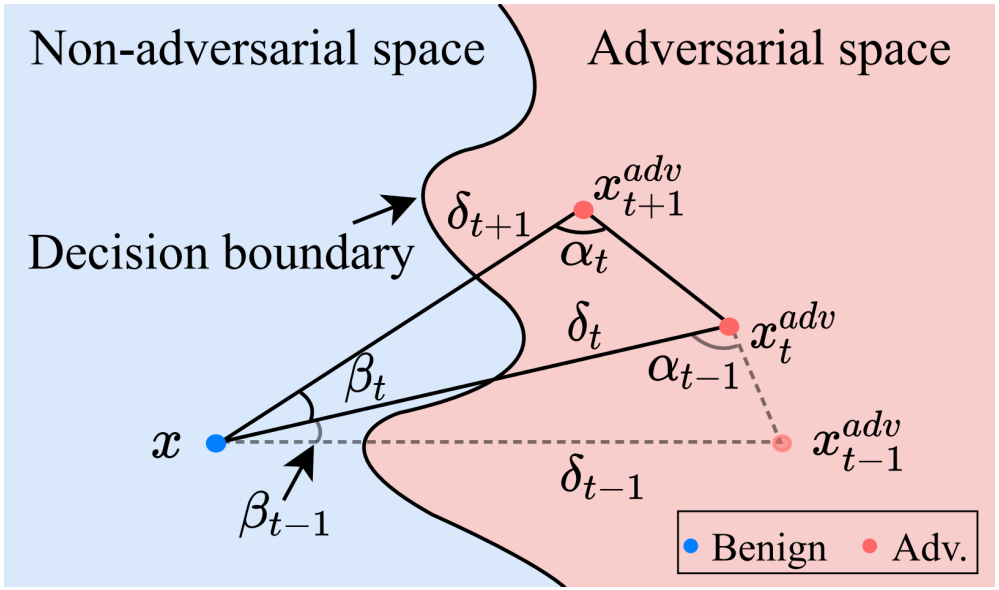
Related works

- **BoundaryAttack** [Brendel et al., 2018] initializes a large perturbation and performs random walks on the decision boundary while keeping adversarial.
- Recent works adopt various gradient estimation strategies to efficiently optimize the perturbation, such as **HSJA** [Jianbo et al., 2020], **QEBA** [Huichen et al., 2020], **GeoDA** [Ali et al., 2020], etc.
- **Surfree** [Thibault et al., 2021]: Iteratively construct a circle on the decision boundary and adopt binary search to find the intersection of the constructed circle and decision boundary as the adversary **without any gradient estimation**.



Triangle Attack

Assumption: Given a benign sample x and a perturbation budget ϵ, there exists an adversarial perturbation $||\text{delta}||_p \leq \epsilon$ towards the decision boundary which can mislead the target classifier f.

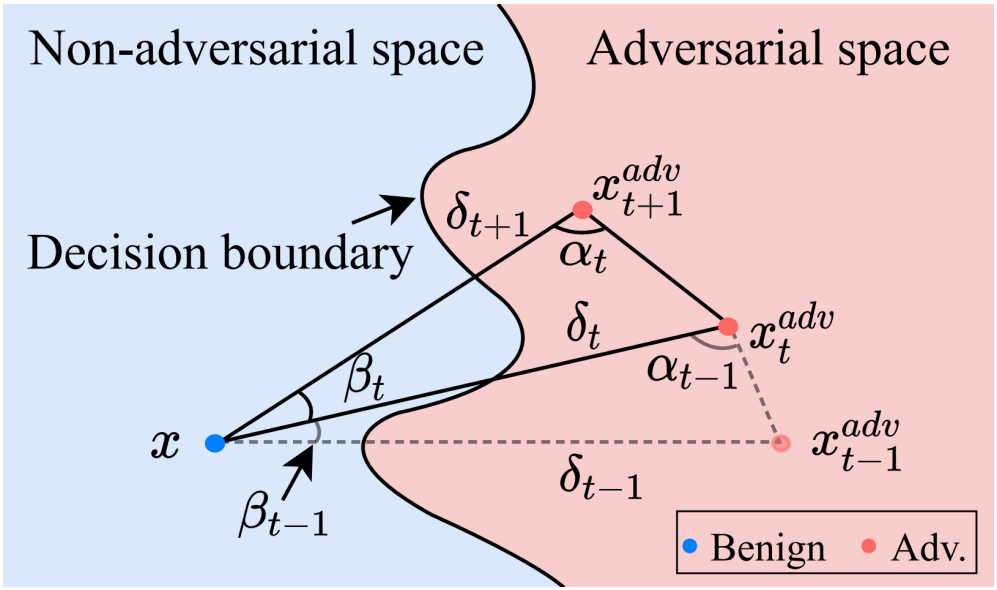


At the t-th iteration, x, x_t^{adv} and x_{t+1}^{adv} can naturally construct a **triangle** for any iterative attacks.

Can we utilize such triangle for efficient attack?

Triangle Attack

Theorem (The law of sines): Suppose a, b and c are the sides lengths of a triangle, and α, β and γ are the opposite angles, we have $\frac{a}{\sin\alpha} = \frac{b}{\sin\beta} = \frac{c}{\sin\gamma}$.

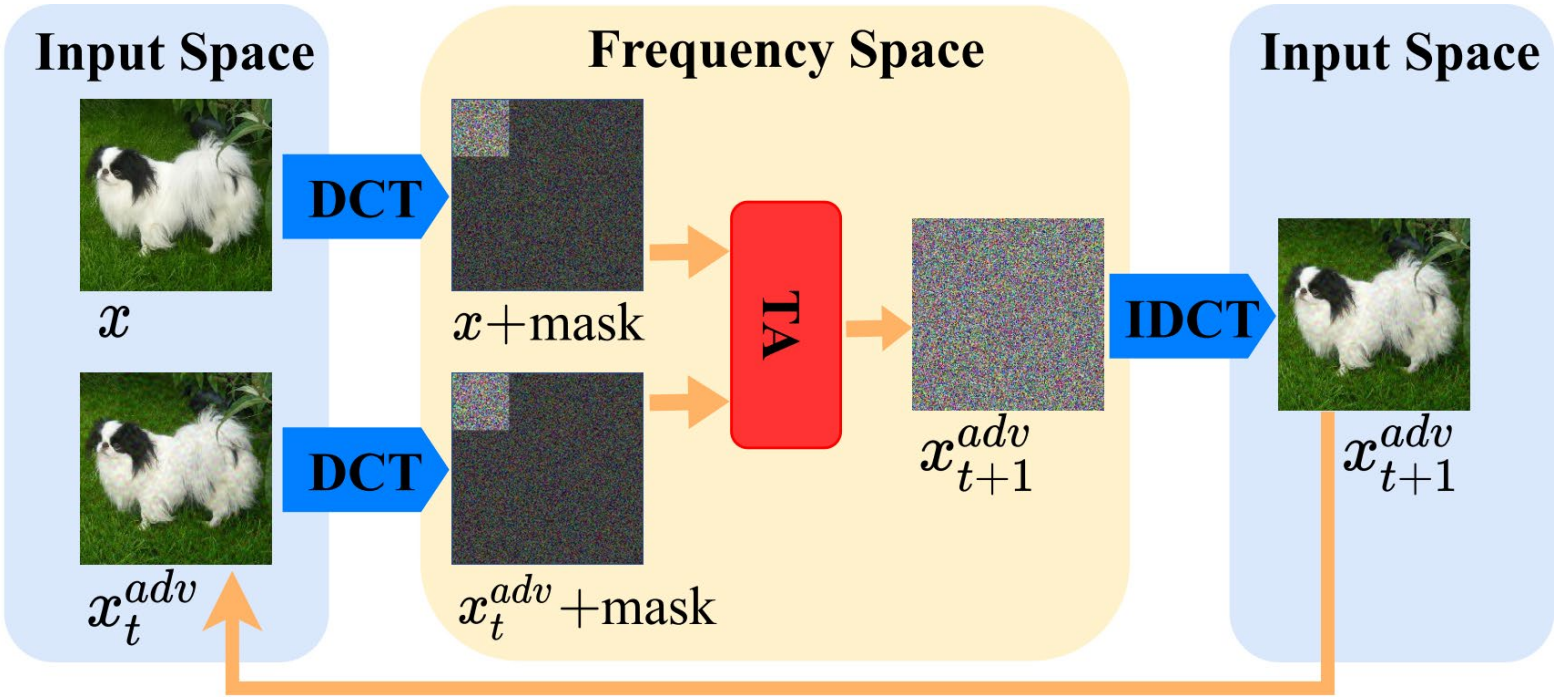


$$\frac{\delta_t}{\sin\alpha_t} = \frac{\delta_{t+1}}{\sin(\pi - (\alpha_t + \beta_t))}$$

To decrease the perturbation ($\delta_t > \delta_{t+1}$), we should guarantee $\pi - (\alpha_t + \beta_t) < \alpha_t$.

Triangle Attack

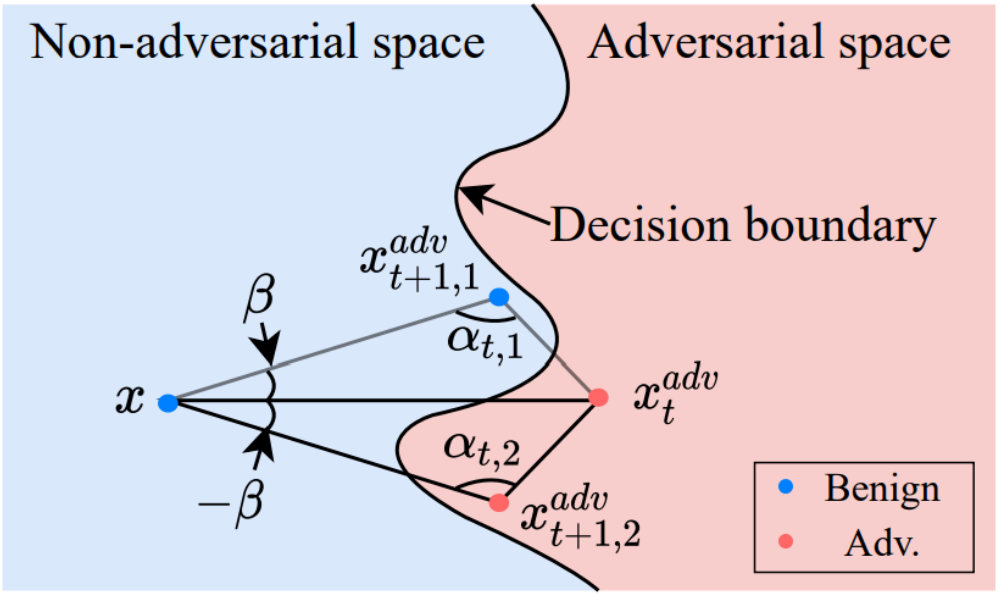
Sampling the 2-D subspace S of frequency space. Thanks to the **generality** of the geometric property, we optimize the perturbation in the **frequency space**.



Triangle Attack

Searching the candidate triangle. x, x_t^{adv}, α and β could determine a triangle to find the next adversarial example x_{t+1}^{adv}. α is a learnable angle and we conduct binary search to find the angle β.

$$\beta^* \in \left[\max\left(\pi - 2\alpha, \frac{\pi}{16}\right), \min\left(\pi - \alpha, \frac{\pi}{2}\right)\right]$$

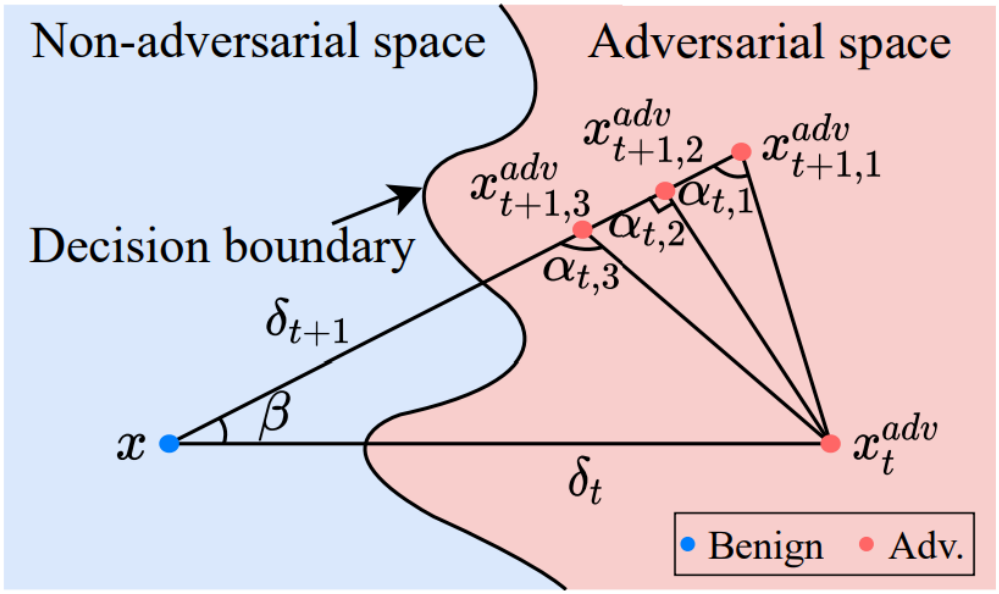


Triangle Attack

Proposition. With the same angle β, a smaller angle α makes it easier to find an adversarial example while a larger angle α leads to smaller perturbation.

Adjusting angle α.

$$\alpha_{t,i+1} = \begin{cases} \min\left(\alpha_{t,i} + \gamma, \frac{\pi}{2} + \tau\right) & if\ adversarial \\ \max\left(\alpha_{t,i} - \lambda\gamma, \frac{\pi}{2} - \tau\right) & Otherwise \end{cases}$$



Triangle Attack

Algorithm 1: Triangle Attack

Input: Target classifier $\boldsymbol{f}$ with parameters $\boldsymbol{\theta}$; Benign sample $\boldsymbol{x}$ with ground-truth label $\boldsymbol{y}$; Maximum number of queries Q; Maximum number of iteration N for each sampled subspace; Dimension of the directional line d; Lower bound $\underline{\beta}$ for angle β.

Output: An adversarial example x^{adv}.

Initialize a large adversarial perturbation δ_0;
$x_0^{adv} = x + \delta_0$, $q = 0$, $t = 0$, $\alpha_0 = \pi/2$;
while $q < Q$ **do**
    Sampling 2-D subspace $\mathcal{S}_t$ in the low frequency space;
    $\beta_{t,0} = \max(\pi - 2\alpha, \underline{\beta})$;
    if $f(\mathcal{T}(x, x_t^{adv}, \alpha_{t,0}, \beta_{t,0}, \mathcal{S}_t); \theta) = f(x; \theta)$ **then**
        $q = q + 1$, update $\alpha_{t,0}$ based on Eq. (3);
        if $f(\mathcal{T}(x, x_t^{adv}, \alpha_{t,0}, -\beta_{t,0}, \mathcal{S}_t); \theta) = f(x; \theta)$ **then**
            $q = q + 1$, update $\alpha_{t,0}$ based on Eq. (3);
            Go to line 3; ▷ give up this subspace
    $\overline{\beta}_{t,0} = \min(\pi/2, \pi - \alpha)$;
    for $i = 0 \to N$ **do** ▷ binary search for angle β
        $\beta_{t,i+1} = (\overline{\beta}_{t,i} + \beta_{t,i})/2$;
        if $f(\mathcal{T}(x, x_t^{adv}, \alpha_{t,i}, \beta_{t,i+1}, \mathcal{S}_t); \theta) = f(x; \theta)$ **then**
            $q = q + 1$, update $\alpha_{t,i}$ based on Eq. (3);
            if $f(\mathcal{T}(x, x_t^{adv}, \alpha_{t,i}, -\beta_{t,i+1}, \mathcal{S}_t); \theta) = f(x; \theta)$ **then**
                $\overline{\beta}_{t,i+1} = \beta_{t,i+1}, \beta_{t,i+1} = \beta_{t,i}$;
        $q = q + 1$, update $\alpha_{t,i+1}$ based on Eq. (3);
    $x_{t+1}^{adv} = \mathcal{T}(x, x_t^{adv}, \alpha_{t,i+1}, \beta_{t,i+1}, \mathcal{S}_t)$, $t = t + 1$;
return x_t^{adv}.

Experimental Settings

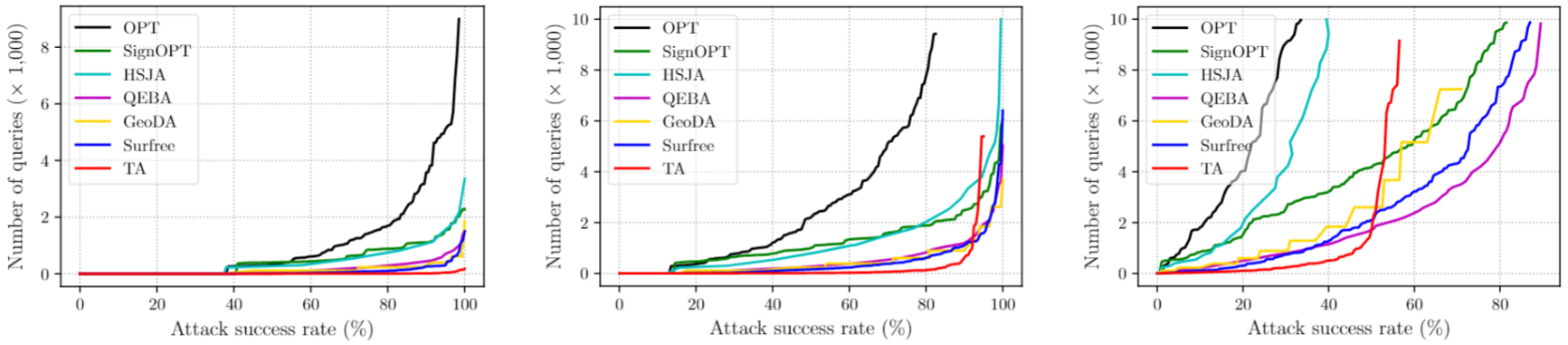
- **Dataset**: ILSVRC 2012 validation set
- **Models**: VGG-16, Inception-v3, ResNet-18, ResNet-101, DenseNet-121 and Tencent Cloud API
- **Baselines**: OPT, SignOPT, HSJA, QEBA, BO, GeoDA, Surfree
- **Evaluation metrics**: RMSE
- **Hyper-parameters**: N=2, d=3, γ=0.01, λ=0.05 and τ=0.1

Experimental Results

Table 1: Attack success rate (%) on five models under different RMSE thresholds. The maximum number of queries is set to 1,000. We highlight the highest attack success rate in **bold**

Model	VGG-16			Inception-v3			ResNet-18			ResNet-101			DenseNet-121		
RMSE	0.1	0.05	0.01	0.1	0.05	0.01	0.1	0.05	0.01	0.1	0.05	0.01	0.1	0.05	0.01
OPT	76.0	38.5	5.5	34.0	17.0	4.0	67.0	36.0	6.0	51.5	21.0	5.0	51.5	29.0	5.5
SignOPT	94.0	57.5	12.5	50.5	27.0	8.0	84.5	49.5	13.0	69.0	33.0	8.0	69.5	44.0	10.0
HSJA	92.5	58.5	13.0	32.5	14.0	4.0	83.0	51.0	12.5	71.5	37.5	12.0	70.5	43.5	10.5
QEBA	98.5	86.0	29.0	78.5	54.5	17.0	98.0	81.5	34.5	94.0	59.0	20.5	91.0	66.0	24.0
BO	96.0	72.5	17.0	75.5	43.0	10.0	94.5	74.0	16.0	89.5	63.0	16.5	93.0	64.5	16.5
GeoDA	99.0	94.0	35.0	89.0	61.5	23.5	99.5	90.0	30.5	98.0	81.5	22.0	**100.0**	84.5	27.5
Surfree	99.5	92.5	39.5	87.5	67.5	24.5	98.5	87.0	36.0	95.5	76.5	27.0	97.0	78.0	29.0
TA (**Ours**)	**100.0**	**95.0**	**44.5**	**96.5**	**81.5**	**30.0**	**100.0**	**94.0**	**51.5**	**99.0**	**88.5**	**40.0**	99.5	**92.5**	**43.5**

Experimental Results



(a) RMSE = 0.1 (b) RMSE = 0.05 (c) RMSE = 0.01

Fig. 5: Number of queries to achieve the given attack success rate on ResNet-18 for the attack baselines and the proposed TA under various perturbation budgets. The maximum number of queries is 10,000

Experimental Results

Table 2: The number of adversarial examples successfully generated by various attack baselines and the proposed TA on Tencent Cloud API within 200/500/1,000 queries. The results are evaluated on 20 randomly sampled images from the correctly classified images in ImageNet due to the high cost of online APIs

RMSE	OPT	SignOPT	HSJA	QEBA	GeoDA	Surfree	TA (**Ours**)
0.1	4/6/6	8/8/9	7/8/8	12/12/12	15/15/15	13/13/13	**17/17/17**
0.05	1/3/3	4/4/7	6/6/8	11/11/12	13/14/14	12/12/13	**15/17/17**
0.01	1/1/2	1/1/3	2/5/6	3/8/9	3/7/12	5/8/10	**8/12/13**

Experimental Results

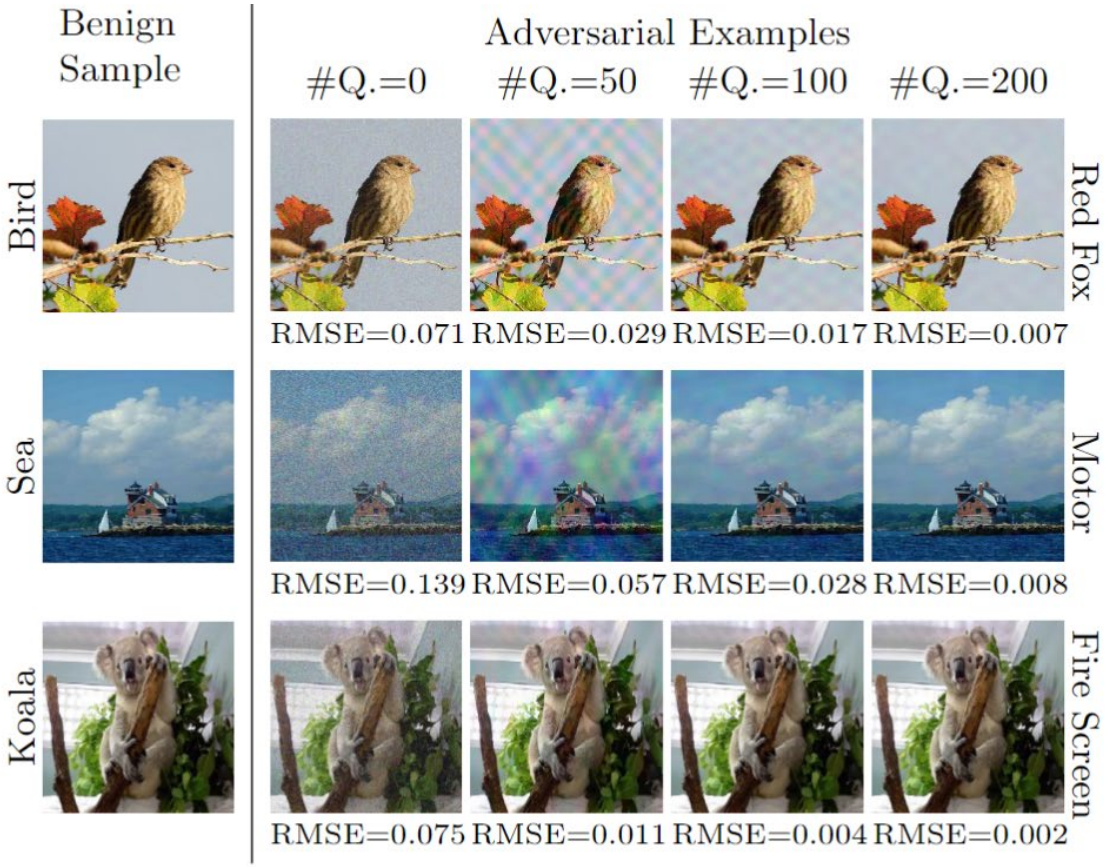


Fig. 6: The adversarial examples crafted by TA against Tencent Cloud API. #Q. denotes the number of queries for attack and RMSE denotes the RMSE distance between the benign sample and adversarial example. We report the correct label and the predicted label on the leftmost and rightmost columns, respectively (Zoom in for details)

Summary

- Propose a novel decision-based attack, called Triangle Attack (TA), which utilizes the **geometric information** that the longer side is opposite the larger angle in any triangle.
- Directly optimizes the adversarial perturbation in the **low frequency space** generated by DCT with much lower dimensions than the input space, and significantly improve the query **efficiency**.
- Achieve a **much higher attack success rate** within 1,000 queries and need **much less queries** to achieve the same attack success rat.
- Demonstrate the **practical applicability** on Tencent Cloud API.

Thanks!

Xiaosen Wang[1,2], Zeliang Zhang[1], Kangheng Tong[1], Dihong Gong[2], Kun He[1], Zhifeng Li[2], Wei Liu[2]

[1] School of Computer Science, Huazhong University of Science and Technology

[2] Data Platform, Tencent

Contact: xiaosen@hust.edu.cn

Homepage: http://xiaosenwang.com/